AD NUMBER

AD888796

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies only; T & E; 8 Nov 71. Other requests shall be referred to ONR, Attn: Code 461, arlington, VA 22217.

AUTHORITY

ONR Ltr., 29 Aug 73

Special Report

PROGRAM BALLOC: A MIXED-WEAPON ALLOCATION MODEL AGAINST MULTIPLE NON-OVERLAPPING AREA DEFENSES

By: R. de SOBRINO B. J. RIPPLE N. J. LEMONS

Distribution limited to U.S. Gov't. agencies only; Test and Evaluation; 8 NOV 1971. Other requests for this document must be referred to

Prepared for:

OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217
Code 461

CONTRACT N00014-71-C-0015

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

PROGRAM BALLOC: A MIXED-WEAPON ALLOCATION MODEL AGAINST MULTIPLE NON-OVERLAPPING AREA DEFENSES

By: R. de SOBRINO B. J. RIPPLE N. J. LEMONS

Prepared for:

OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-71-C-0015

SRI Project 8777

Approved by:

I. J. WEINSTEIN, *Director*
Systems Evaluation Department

E. J. MOORE, *Executive Director*
Engineering Systems Division

Copy No. 32

ABSTRACT

Program BALLOC is a model designed to optimally allocate a mixed-weapon offensive force over a set of multiple-valued targets defended by multiple nonintersecting random area and subtractive local defenses.

The program will maximize the expected damage over the total target set, with the restriction that no more than one weapon type be allocated to any randomly defended island or target subset.

This report presents the concept on which the model is based, and describes the implementation of this concept as a computer program. It also contains a users manual, and an example of the use of the model, employing fictitious data to avoid the necessity of security classification.

PREFACE

This report is submitted under Contract N00014-71-C-0015 to the Office of Naval Research under the task, "Programming a Strategic Bomber Allocation Model." This report contains a description of the model as well as a users guide for running the program.

The work reported here has been performed in the Systems Evaluation Department, Engineering Systems Division of Stanford Research Institute, under the supervision of Dr. Ricardo de Sobrino, Staff Scientist, Mrs. Barbara J. Ripple, Mathematician, acted as project leader, and Mrs. Nancy J. Lemons, Mathematician, did the programming.

CONTENTS

ABSTRACT. ii

PREFACE iii

LIST OF ILLUSTRATIONS vi

I INTRODUCTION AND PURPOSE 1

II MODEL DESCRIPTION. 2

A. General 2

B. Problem Formulation 3

III MODEL IMPLEMENTATION 7

A. Introduction. 7

B. Generating the Corridor Damage Curves 7

1. General. 7
2. Finding Penetration Probabilities. 8
3. Target Damage Functions. 9
4. Find the Desired Attack Size 10

C. Optimal Weapon Allocation 11

D. Closing Procedure 13

IV PROGRAM DESCRIPTION. 15

A. Introduction. 15

B. Program Flow. 16

C. Data Input. 16

1. Data-Card Deck Set-up. 16
2. Input-Card Formats 20

D. Sample Output 22

Appendix--PROGRAM LISTING 37

REFERENCES. 56

DD Form 1473

ILLUSTRATIONS

Figure 1 BALLOC Flow Chart 17

Figure 2 Allocation Subroutine 18

Figure 3 Closing Subroutine. 19

Figure 4 Variable-Size Card. 23

Figure 5 Damage-Curve-Point Limits 23

Figure 6 Damage-Curve-Size Lirits. 24

Figure 7 λ Adjustment Card 24

Figure 8 "K" Value Card. 25

Figure 9 Local-Defense Card. 25

Figure 10 Targets-per-Corridor Card 26

Figure 11 Target-Description Card 26

Figure 12 Corridor-Entry-Point Card 27

Figure 13 Interceptor Constants 27

Figure 14 Interceptor Reliability 28

Figure 15 Attack-Size Table 28

Figure 16 Distance Table. 29

Figure 17 Penetration-Probability Table 29

Figure 18 Desired-Allocation Card 30

Figure 19 Program Input 31

Figure 20 Corridor Damage Functions 32

Figure 21 Iteration Summary 34

Figure 22 Final Corridor Allocation 35

Figure 23 Summary 36

I INTRODUCTION AND PURPOSE

During the past few years, models used to evaluate the effectiveness of strategic systems have been highly developed and widely used. In particular, one general class of such models, which will be called here the "aggregated type"* has become attractive in the rapid evaluation of relative effectiveness of alternative systems. In addition, these models are useful in investigating the sensitivity of the results to the values of system parameters, and to the interactions of offense and defense.

SRI is currently under contract with OSD(SA) to develop techniques to improve the representation of airborne strategic systems in effectiveness models of the aggregated type. As a result of that work a weapons-allocation model against targets defended by random type area defenses with certain restrictions has been made possible.

The purpose of this report is to describe BALLOC, a weapons-allocation model programmed for the Office of the Chief of Naval Operations.

---

*For a more complete discussion, see Ref. 1. References are listed at the end of the report.

II MODEL DESCRIPTION

A. General

The allocation model described here has some unique features that are not available in currently existing models of the aggregated type, such as CODE 50, AEM, and similar ones.

The model assumes a number of corridors (or islands) defended by area defenses (fighters) that do not overlap. It also assumes that no more than one type of bomber (more specifically, payload type) will be used within each island. These are the main limitations. It does not appear easy to eliminate the first. Methods to remove the second are well under development,[2] though this can only be done at the expense of increased running time.

Distinctive features of this model are as follows:

(1) Bomber penetration probabilities are a function of the attack size on an island (or corridor).

(2) They also vary within each island as a function of the penetration distance; in fact, they may be different for each individual target.

(3) They are accepted as inputs in tabular or functional form and may be obtained from detailed simulations if desired.

B. Problem Formulation

We assume here that a set of targets is defended by non-overlapping multiple random fighter area defenses. Since the defense coverages are nonoverlapping, the targets are in fact divided into N disjoint subsets (or corridors). Each corridor is defended by a known random fighter defense. In addition, subtractive local defense may also exist for some (or all) targets in the corridor. We further assume that each corridor will be attacked by at most one type of weapon.

We should note that the model allocates the weapons, not the carriers. This self-imposed restriction is needed to achieve acceptable computation times, and is generally assumed in models of this type. Once a weapon allocation is obtained, weapons could, in the real case, be loaded on specific carriers that could then be routed to specific targets.

Let B_{ij} be the number of type j weapons allocated to the i^{th} corridor. The penetration probabilities of the weapons are given inputs in tabular form, $P_{ij}(B_{ij}, R)$, where R is the distance from the entry point to each island. Thus, and in general, penetration probabilities are different for different targets. Then for a given value of B_{ij}, the penetration probability of each weapon can be obtained from the prescribed penetration probability function $P_{ij}(B_{ij})$, which in turn determines the damage function of each target; the optimal allocation of B_{ij} that maximizes the total expected damage function of the island can be efficiently generated by the use of Lagrange multipliers.

Let $f_{ij}(B_{ij})$ be the total expected damage function to the i^{th} island caused by type j weapons. We note that f_{ij} is a function in tabular form since it corresponds to the sum of the expected damages of all the targets in the i^{th} corridor with each target having in general a different damage function, because the targets either belong to different exemplar groups, or have different penetration probabilities due to geographic location.

The problem may be expressed mathematically as follows:

$$\underset{(U_{ij})\,(B_{ij})}{\text{Maximize}} \sum_{i=1}^{N} \sum_{j=1}^{J} U_{ij}\, f_{ij}\left(B_{ij}\right)$$

$$\text{Subject to} \quad \sum_{j=1}^{J} U_{ij} \le 1 \quad , \quad i = 1,2,\ldots,N$$

$$\sum_{i=1}^{N} B_{ij} \le B_j \quad , \quad j = 1,2,\ldots,J$$

$$B_{ij} \ge 0 \text{ for all } i,j$$

$$U_{ij} = 0 \text{ or } 1 \text{ for all } i,j$$

where J is the total number of weapon types, and U_{ij} equals 1 if the type j weapon is assigned to the i^{th} corridor, and equals 0 otherwise.

The generalized Lagrange multiplier method is readily applicable to this problem. The method consists of two parts:

(1) For a given set of values of the Lagrange multipliers $\lambda = (\lambda_1, \cdots, \lambda_J)$, let $B^0 = (B_{ij}^0)$ be the weapon allocation matrix that maximizes the Lagrangian:

$$L(\lambda, B) = \sum_{i=1}^{N} \sum_{j=1}^{J} \left[f_{ij}\left(B_{ij}\right) - \lambda_j B_{ij} \right] = \sum_{i=1}^{N} \sum_{j=1}^{J} L_{ij}\left(\lambda_j, B_{ij}\right) \quad .$$

For the i^{th} corridor, let j_i be the weapon type that maximizes

$$L_{ij}\left(\lambda_j, B_{ij}^0\right), \text{ i.e.,}$$

$$L_{ij_i}\left(\lambda_j, B_{ij_i}^0\right) = \max L_{ij}\left(\lambda_j, B_{ij}^0\right) \quad .$$

Let $U_{ij}^0 = 1$ if $j = j_i$

$= 0$ otherwise,

i.e., assign weapons of type j_i to the i^{th} corridor and let

$$B_j^* = \sum_{i=1}^{N} U_{ij}^{\lambda} B_{ij}^{\lambda} \qquad .$$

Then U_{ij}^{λ} and B_{ij}^{λ} are optimal for the assignment problem, with B_j^* in place of B_j for the total number of type j weapons.

(2) Using an existing iterative procedure, the values of λ. may be systematically adjusted so that the weapon resource vector (B_J^*) converges to its desired values (B_j).

III MODEL IMPLEMENTATION

A. Introduction

The program consists basically of three steps: (1) finding the damage functions for each attacker type on each corridor, (2) optimally allocating the offense force on the corridors, and (3) adjusting the optimal allocation to be identical with the desired allocation when the latter is not exactly obtained. This last step will be denoted by "closing."

B. Generating the Corridor Damage Curves

1. General

Because the probability of bomber penetration to each target (and the corresponding damage function) depends on the total attack size to a corridor, due to the assumption of random area defenses, the following steps are necessary to obtain each point on the damage curve of a given weapon type on a given corridor:

(1) For each desired attack size, find the penetration probability corresponding to that attack size.

(2) Find the individual damage functions for each target given the penetration probability.

(3) Using Lagrange multipliers, find the optimal allocation for the desired attack size.

Each of these three steps is discussed in more detail below.

The number of points on each curve, as well as the maximum attack size for each curve, are user inputs. Due to computer storage requirements, however, a maximum of 20 points per curve cannot be presently exceeded.

2. Finding Penetration Probabilities

To find the probability of a weapon penetrating to a target, t, the great-circle distance, d, from the corridor entrance to that target is found using:

$$c = \cos^{-1} [\sin(LA_{p1}) \sin(LA_T) + \cos(LA_{p1}) \cos(LA_T) \cos(LO_T - LO_{p1})]$$

$$e = \cos^{-1} [\sin(LA_{p1}) \sin(LA_{p2}) + \cos(LA_{p1}) \cos(LA_{p2}) \cos(LO_{p2} - LO_{p1})]$$

$$B = \cos^{-1} [\sin(LA_T) - \cos(c) \sin(LA_{p1})] / [\sin(c) \cos(LA_{p1})] -$$

$$\cos^{-1} [\sin(LA_{p2}) - \cos(e) \sin(LA_{p1})] / [\sin(e) \cos(LA_{p1})]$$

$$d = 3400 \sin^{-1} [\sin(c) \sin(B)] \qquad (1)$$

where

LA_T, LO_T = Latitude and Longitude of target t

LA_{p1}, LO_{p1}, LA_{p2}, LO_{p2} = Latitudes and Longitudes of corridor entry boundaries

d = Great-circle distance between target t and corridor entry line defined by p1 and p2.

Given this distance and the attack size of interest, a table lookup procedure is used to obtain the penetration probability. These tables are inputs that must be generated externally.[5] Linear interpolation is used to obtain the desired attack size from the tables.

3. Target Damage Functions

The square-root-law damage function is used for each target. For a locally undefended target with penetration probability equal to one, this is of the following form:

$$\theta_i(a_{j,i}) = v_i\left[1 - e^{-k_{j,i}\sqrt{a_{j,i}}}\,(1 + k_{j,i}\sqrt{a_{j,i}})\right] \qquad (2)$$

where

v_i = Target value

$a_{j,i}$ = Number of type j weapons arriving at target i

$k_{j,i}$ = A damage constant that depends on the target's hardness and size and the weapon's yield and CEP

θ_i = Expected damage.

Given a penetration probability (P_{ij}) for attacker j on target i, a new value of k, k′ is found such that

$$P_{i,j}\theta_i(1) = v_i\left[1 - e^{-k'_{j,i}}(1 + k'_{j,i})\right] \quad . \qquad (3)$$

This k' value then has the effect of "stretching" the undefended damage function to allow for the effects of the penetration probability through the area defenses, and the expected damage is found by using this k' value in Eq. (2).

If there is a local defense at target i, the $\sqrt{a_{j,i}}$ term in Eq. (2) is replaced by $\sqrt{a_{j,i} - Y_{j,i}}$, where

$$Y_{j,i} = I_{j,i} R_j k P_{j,i} \tag{4}$$

and

$Y_{j,i}$ = Equivalent price of the target i for weapon type j

$I_{j,i}$ = Interceptors at target i effective against weapon j

R_j = Interceptor kill probability against weapon type j

k = An input constant associated with the leakage through the area defenses[a]

$P_{j,i}$ = Penetration probability through the area defenses, a function of the target location and attack size.

4. Find the Desired Attack Size

Given the individual target damage functions at a corridor, it is then possible to obtain the attack size of interest. This is done using Lagrange multipliers in an iterative technique. Starting with a very large value of a single

scalar multiplier and a very small one, two attack levels that bracket the desired attack size are obtained. A new value for the multiplier is found by taking the average of the two previous ones. The new allocation is obtained and replaces the old point on the same side of the desired allocation. This averaging procedure continues until either the desired point is found or the values of the multipliers do not coalesce into a single value resulting in the desired number of weapons, in which case the solution is in a "gap." When this occurs, the point closest to the desired point is used.

C. Optimal Weapon Allocation

To optimally allocate the offensive force, (i.e., maximize the expected damage destroyed over all the corridors) the program again utilizes the Lagrange multiplier method.[4] In this case, the Lagrange multiplier is a vector in j space where j is the number of weapon types.

Given the multiplier vector, the problem of finding the optimum force allocation is quite simple, since the damage functions are in tabular form and the corridors do not overlap. Each corridor can be treated independently, and the "best" weapon type (and the number of weapons of this type) to use on the corridor is found by searching each damage table to find the point that maximizes $F(A) - \lambda_j A = L_j$ for payload type j, and then choosing the type with the largest L_j.

After each allocation, the multiplier vector is adjusted until either the desired attack size (to within the granularity of the damage table) is obtained, or, the change in the values of the multipliers is very small, or a maximum number of iteration is reached. If the desired attack size is obtained, the run is complete; otherwise, a closing procedure is required to adjust the optimal attacks to match the desired one.

To adjust the multiplier vector, the following standard algorithm is used.* Along with the λ_j associated with each attacking payload type is a $\Delta\lambda_j$. It is the $\Delta\lambda_j$ that is adjusted to find the new λ_j. If $\Delta\lambda_j'$ is going in the right direction but the obtained allocation is still not at the desired value, then $\Delta\lambda_j' = c_1 \Delta\lambda_j'$, where c_1 is an input constant > 1.0. If in the last iteration the desired allocation was crossed, the $\Delta\lambda_j' = c_2 \Delta\lambda_j$, where c_2 is also an input constant, negative, and < 1.0. Also, a Δx term is applied to all λ_j's. This term is modified at each iteration by the same constants as the $\Delta\lambda_j$'s, depending on whether $\sum_j \lambda_j W_D$ is greater or less than $\sum_j \lambda_j W_A$ (W_D = desired allocation; W_A = obtained allocation). Thus,

$$\lambda_j' = (1 + \Delta\lambda_j')\ (1 + \Delta x)\ \lambda_j \quad .$$

The values of c_1 and c_2 have been obtained heuristically.

---

* This algorithm was originated by Dr. H. Everett of Lambda Corp.

D. Closing Procedure

The closing procedure is a heuristic method used to slightly change an obtained optimal allocation to reach the desired one when the latter lies in a "gap" and is not obtainable by the Lagrange multiplier method.

In Program BALLOC, two procedures are needed; the first adjusts the allocation among the corridors until the desired allocation is obtained. The second procedure then adjusts the individual corridor allocations among the targets within that corridor.

The first step in performing the first closing is to select the "best" allocation to close on.[5] This is done by finding the allocation that minimizes:

$$\Sigma_j \left[\left(W_{A_j} - W_{D_j}\right) / W_{D_j}\right]^2 \tag{5}$$

where

W_{A_j} = Obtained number of type j weapons

W_{D_j} = Desired number of type j weapons.

The routine imposes the restriction that each corridor must use the weapon type assigned in the optimal allocation; thus, for each type-j weapon, if $W_{A_j} > W_{D_j}$, the routine selects from the corridors that are assigned this type the one receiving

the least damage per attacker, and reduces the allocation to this corridor. Similarly, if $W_{A_j} < W_{D_j}$, the corridor that will receive the most damage per attacker is selected to receive the additional weapons.

If a weapon type was not allocated at all, the routine selects from the unattacked corridors the one that would receive the most damage and allocates the weapons to that corridor. After the exact allocation to each corridor is obtained, the procedures described previously in Section B-4 are used to allocate the weapons among the targets within the corridor. If the desired allocation to that corridor is not exactly obtained, an algorithm similar to that used in the first closing is used for slight adjustments.

IV PROGRAM DESCRIPTION

A. Introduction

BALLOC is written in Fortran IV for use on the SRI CDC 6400 computer as well as the CNA CDC 3600. The core storage requirement is approximately 50,000 words. The running time can be broken into two parts--the time required to generate the damage curves for each corridor, and, given the curves, the time required to obtain a given allocation. To generate a curve with 20 points for one weapon type on one corridor requires approximately 10 seconds on the CDC 6400.* The running time to obtain the desired weapon allocation depends on the number of corridors and the number of weapon types. For two weapon types and nine corridors, the time is approximately 5 seconds.

Section IV-B contains flow charts of the main program and of the more complicated subroutines. Section IV-C gives a description of the user input to the program, along with typical sample values. In Section IV-D are several figures showing various representative sections of the output of this program.

The target values and damage functions used to calculate the examples shown are fictitious to avoid security classification.

---

*This is an average value assuming 40 targets per corridor and obtained from a total time of about 180 sec for 400 targets located in 9 corridors.

B. Program Flow

Figure 1 is a flow chart of the main program flow. Figure 2 shows the allocation subroutine, and Figure 3 the closing procedure.

C. Data Input

1. Data-Card Deck Set-up

The following cards are required to run the program:

Card ID	Number of Cards	Description
1	1	Variable size card
2	1	Number of points on damage curves for each weapon type
3	1	Maximum number of weapons of each type to be used in computing damage curves
4	1	λ adjustment card
5	A set for each weapon type	"K" values for each exemplar group
6	One for each locally defended target	Local-defense card
7	1	Number of targets in each corridor
8	One for each target	Target-description card*
9	One for each corridor	Corridor entry points
10	1	Interceptor constants

* These cards are ordered so that all Corridor -1 cards are first, followed by all Corridor-2 cards, etc.

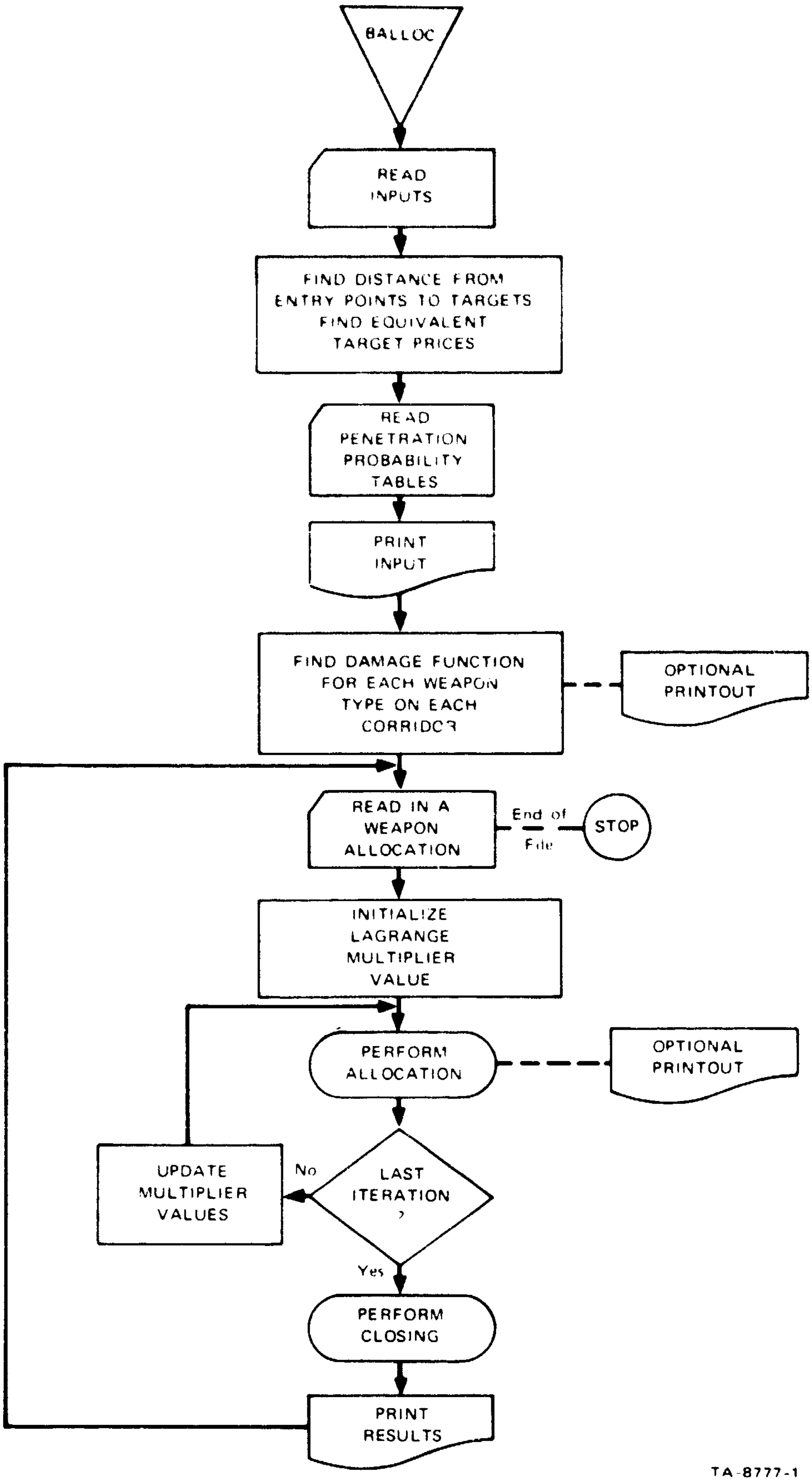

FIGURE 1 BALLOC FLOW CHART

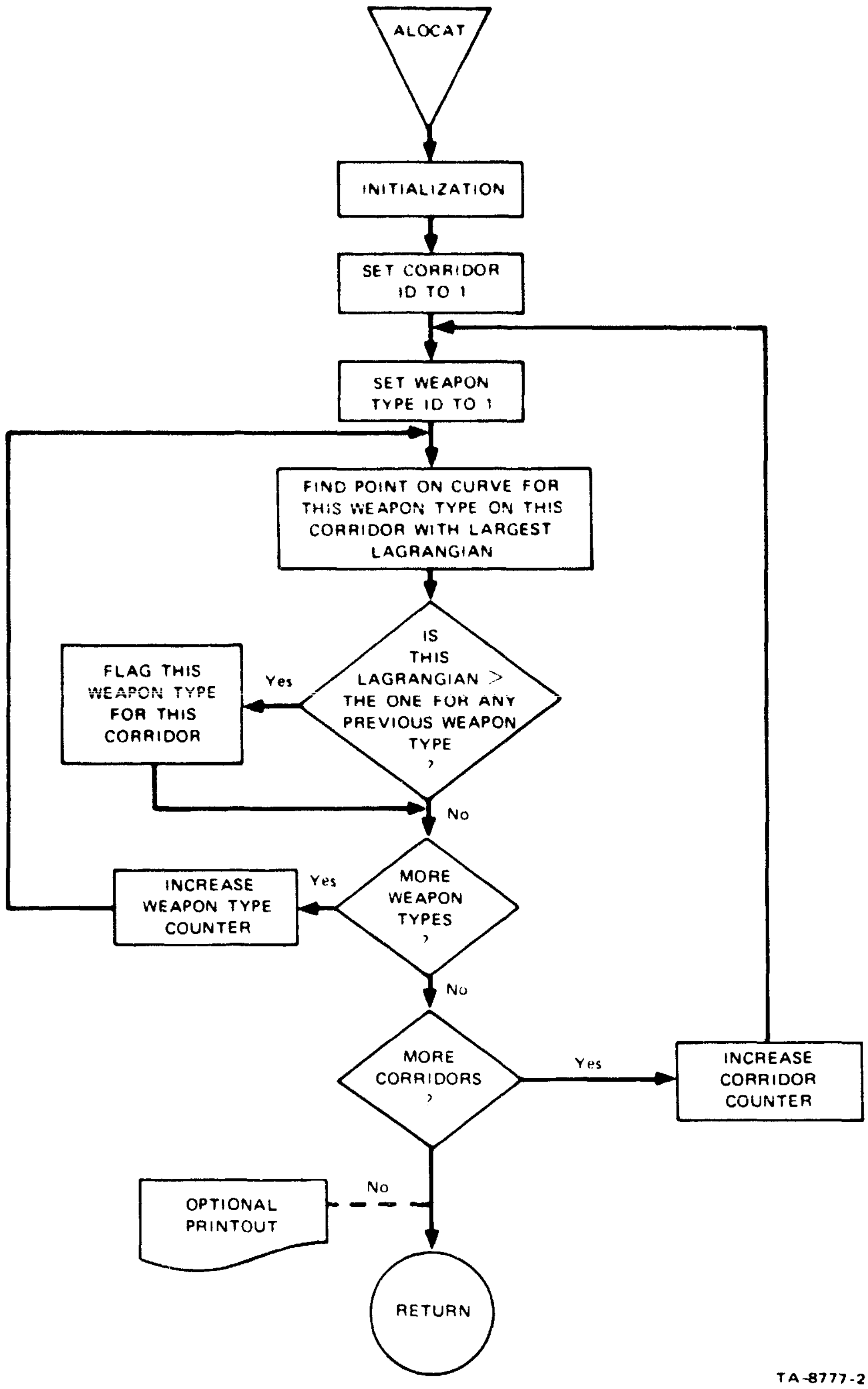

FIGURE 2 ALLOCATION SUBROUTINE

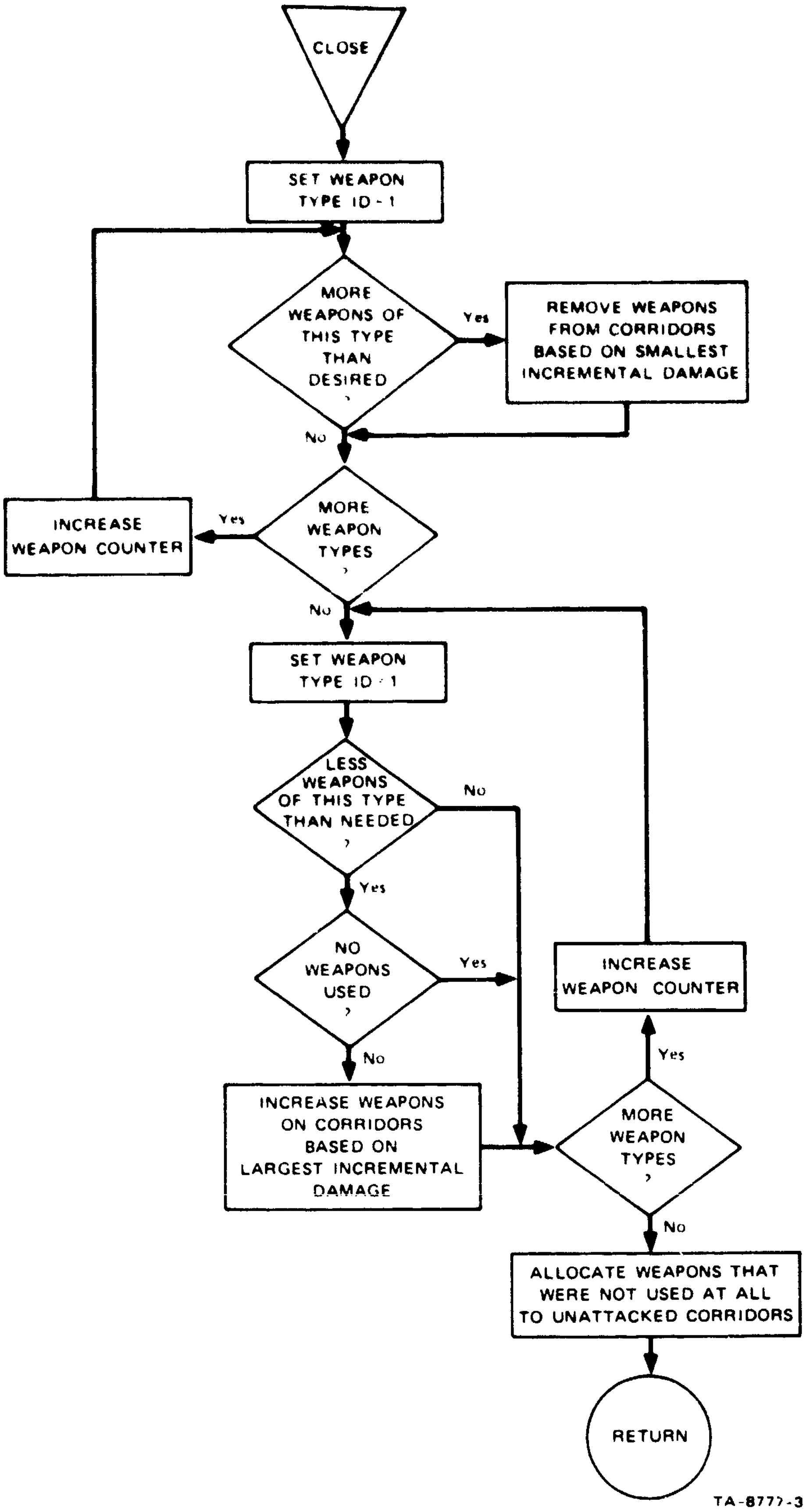

FIGURE 3 CLOSING SUBROUTINE

Card ID	Number of Cards	Description
11	1	Interceptor reliability against each weapon type
12*	1	Attack-size table
13*	As many as needed	Distance from entry-line table
14*	As many as needed	Penetration probabilities as a function of the attack size and distance
15	One for each case to be run	Desired-allocation card

2. Input-Card Formats

Figures 4 through 18 describe the input card formats. The first row contains the number type (I = right justified integer, R = a real number, and A is an alpha format). The second row contains the field on which the number is to appear. The third row describes the input, and the fourth row gives sample values.

Figure 4 contains the size limitations for the various parameters as well as giving three print options. If the Option 1 field is not blank or 0, damage curves for each weapon type on each island will be plotted. Option 2 will print out the allocation to the corridors for each iteration, unless the field is blank or 0. Similarly, a blank or 0 for Option 3 will suppress the printing of the allocation summary for each cycle.

Figure 7 contains the constants for increasing or decreasing the Lagrange multiplier vector to converge on the desired solution, as well as the cutoff point to stop the iterations.

*These are inserted in the above order for each corridor.

Each target is assigned to an exemplar group, however, there may be up to 400 groups so that each target can be treated independently. The damage constant ("k") (Figure 8) is given for each weapon type and each exemplar group.

The target cards (Figure 11) are ordered so that all of the targets in Corridor 1 are first, followed by Corridor 2, etc. Figure 10 gives the number of targets in each corridor. The last corridor contains all of the targets not covered by any area defense.

For each city with local defense, a card (Figure 9) showing the number of interceptors against each weapon type is used. Also, a constant determined by the leakage through the area defense is needed for each weapon type (Figure 13). Interceptor kill probabilities are given against each weapon type (Figure 14), and to obtain the equivalent intercepts of weapon j against target i the program uses

$$E_{ij} = I_{ij} R_j K_j \frac{1}{P_p}$$

where

I_{ij} = Interceptors at target i against weapon type j

R_j = Interceptor kill probability against weapon type j

K_j = An input constant associated with the leakage through the area defenses.[3]

P_p, the penetration probability to a given target, is a function of the total attack size to the corridor and the distance of the target from the corridor entrance. Figures 16, 17, and 18 contain the tables describing these functions.

D. Sample Output

Figures 19 through 23 are samples of the output available from BALLOC. Figure 19 contains the program inputs. Figure 20 shows the plots of the individual damage functions for each weapon type on each corridor. This printout is optional, as is Figure 21, which contains the output available after each iteration. Figures 22 and 23 show the final program output.

I5	I5	I5	I5	I5	I5	I5	[illegible]	[illegible]	[illegible]
1 5	6 10	11 15	16 20	21 25	26 30	31	[illegible]	[illegible]	[illegible]
number of corridors	number of attacker types	p_p table: number of attack sizes	p_p table: number of distances	maximum iterations	print option 1	print option [illegible]	print [illegible]	[illegible]	[illegible]
10 (maximum of 15)	5 (maximum of [illegible])	[illegible] (maximum of [illegible])	12 (maximum of 12)	20	1	[illegible]	[illegible]	[illegible]	[illegible]

FIGURE 4 VARIABLE-SIZE CARD

I5	I5	I5	
1 5	6 10	11 15	• • • up to 5 weapon types
number of points to be used in damage curves for weapon 1	number of points to be used in damage curves for weapon 2	number of points to be used in damage curves for weapon 3	
15 (maximum of 20)	20	12	

FIGURE 5 DAMAGE-CURVE-POINT LIMITS

I5	I5	I5	
1 5	6 10	11 15	• • • up to 5 weapon types
maximum number of type 1 weapons to be used in calculating damage curves	maximum number of type 2 weapons to be used in calculating damage curves	maximum number of type 3 weapons to be used in calculating damage curves	
100	450	35	

FIGURE 6 DAMAGE-CURVE-SIZE LIMITS

R	R	R
1 10	11 20	21 30
$\Delta\lambda$ decrease constant	$\Delta\lambda$ increase constant	$\Delta\lambda$ cutoff
-0.25	1.3	0.005

FIGURE 7 λ ADJUSTMENT CARD

R	R	R	
1 — 10	11 — 20	21 — 30	• • • up to 400 exemplar groups
"K" for exemplar group 1	"K" for exemplar group 2	"K" for exemplar group 3	
1.2	0.8	2.5	

FIGURE 8 "K" VALUE CARD

I5	I5	I5	I5	
1 — 5	6 — 10	11 — 15	16 — 20	• • • up to 5 weapon types
target ID number	interceptors against weapon type 1	interceptors against weapon type 2	interceptors against weapon type 3	
5	10	8	10	

FIGURE 9 LOCAL-DEFENSE CARD

I5	I5	I5	
1 5	6 10	11 15	• • • up to 15 corridors
number of targets in corridor 1	number of targets in corridor 2	number of targets in corridor 3	
10	15	8	

FIGURE 10 TARGETS-PER-CORRIDOR CARD

I10	I10	R	I3	I2	I2	I3	I2	I2
1 10	11 20	21 30	31 36	37 38	39 40	41 46	47 48	49 50
target ID	exemplar group	value	latitude degrees	latitude minutes	latitude seconds	longitude degrees	longitude minutes	longitude seconds
1	3	1000.0	56	30	15	120	27	32

FIGURE 11 TARGET-DESCRIPTION CARD

I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2
1 6	7 8	9 10	11 16	17 18	19 20	21 26	27 28	29 30	31 36	[illegible]	[illegible]
latitude of first entry point boundary (deg.)	(min.)	(sec.)	longitude of first entry point boundary (deg.)	(min.)	(sec.)	latitude of second entry point boundary (deg.)	(min.)	(sec.)	longitude of second entry point boundary (deg.)	(min.)	(sec.)
71	1[illegible]	30	65	10	[illegible]	70	1	[illegible]	80	1[illegible]	1[illegible]

FIGURE 12 CORRIDOR-ENTRY-POINT CARD

R	R	R	
1 10	11 20	21 30	• • • up to 5 weapon types
interceptor constant for weapon type 1	interceptor constant for weapon type 2	interceptor constant for weapon type 3	
1.0	1.2	1.6	

FIGURE 13 INTERCEPTOR CONSTANTS

R	R	R	
1 10	11 20	21 30	• • • up to 5 attacker types
reliability against attacker type 1	reliability against attacker type 2	reliability against attacker type 3	
0.9	0.75	0.8	

FIGURE 14 INTERCEPTOR RELIABILITY

R	R	R	
1 10	11 20	21 30	• • • up to 5 attack levels
first attack level	second attack level	third attack level	
10.0	20.0	50.0	

FIGURE 15 ATTACK-SIZE TABLE

R	R	R	
1 10	11 20	21 30	• • • up to 12 distances
first distance cutoff-nmi	second distance cutoff	third distance cutoff	
50.0	100.0	150.0	

FIGURE 16 DISTANCE TABLE

R	R	R	
1 10	11 20	21 30	• • • up to 12 entries
penetration probability for first distance cutoff	P_p for second cutoff	P_p for third cutoff	
0.8	0.7	0.6	

FIGURE 17 PENETRATION-PROBABILITY TABLE

R	R	R	
1 10	11 20	21 30	• • • up to 5 weapon types
desired number of weapon type 1	desired number of weapon type 2	desired number of weapon type 3	
50.0	100.0	50.0	

FIGURE 18 DESIRED-ALLOCATION CARD

CORRIDOR NUMBER 5

ID	VALUE	DIST	K					INTERCEPTORS				
4	2000.00	875.71	.5000	.7500	0.0000	0.0000	0.0000	2	3	0	0	0
10	1000.00	1323.74	.5000	.8000	0.0000	0.0000	0.0000	0	0	0	0	0
15	1000.00	1166.25	.5000	.7500	0.0000	0.0000	0.0000	0	0	0	0	0
43	600.00	807.66	.5500	.9000	0.0000	0.0000	0.0000	0	0	0	0	0
47	500.00	1061.54	.6000	1.0000	0.0000	0.0000	0.0000	0	0	0	0	0
51	500.00	63.44	.6000	1.0000	0.0000	0.0000	0.0000	0	0	0	0	0
61	400.00	1487.91	.6500	1.0000	0.0000	0.0000	0.0000	0	0	0	0	0
67	400.00	1645.90	.6500	1.0500	0.0000	0.0000	0.0000	0	0	0	0	0
73	400.00	369.32	.6500	1.0500	0.0000	0.0000	0.0000	0	0	0	0	0
84	300.00	1007.05	.6500	1.0500	0.0000	0.0000	0.0000	0	0	0	0	0
90	300.00	761.98	.6500	1.0500	0.0000	0.0000	0.0000	0	0	0	0	0
98	300.00	1684.77	.5000	.8000	0.0000	0.0000	0.0000	0	0	0	0	0
130	200.00	1809.03	.8000	1.3000	0.0000	0.0000	0.0000	0	0	0	0	0
142	200.00	670.17	.8000	1.3000	0.0000	0.0000	0.0000	0	0	0	0	0
156	200.00	361.60	.7500	1.2000	0.0000	0.0000	0.0000	0	0	0	0	0
190	200.00	852.30	.7000	1.1000	0.0000	0.0000	0.0000	0	0	0	0	0
210	100.00	899.10	1.0000	1.6000	0.0000	0.0000	0.0000	0	0	0	0	0
256	100.00	852.04	.9000	1.4500	0.0000	0.0000	0.0000	0	0	0	0	0
259	100.00	790.42	.9000	1.4500	0.0000	0.0000	0.0000	0	0	0	0	0
260	100.00	913.30	.9000	1.4500	0.0000	0.0000	0.0000	0	0	0	0	0
267	100.00	1246.32	.9000	1.4500	0.0000	0.0000	0.0000	0	0	0	0	0
277	100.00	1813.28	1.0500	1.7500	0.0000	0.0000	0.0000	0	0	0	0	0
284	100.00	804.01	1.0500	1.7500	0.0000	0.0000	0.0000	0	0	0	0	0
285	100.00	1147.02	1.0500	1.7500	0.0000	0.0000	0.0000	0	0	0	0	0
301	100.00	1079.68	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
326	100.00	1143.87	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
331	100.00	922.38	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
338	100.00	1460.95	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
343	100.00	124.01	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
344	100.00	883.21	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
349	100.00	1742.60	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
350	100.00	1658.42	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
354	100.00	1648.98	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
355	100.00	818.67	1.2000	1.9500	0.0000	0.0000	0.0000	0	0	0	0	0
363	60.00	979.88	.2500	.4000	0.0000	0.0000	0.0000	0	0	0	0	0

FIGURE 19 PROGRAM INPUT

CORRIDOR NUMBER 7
WEAPON NUMBER 1

DAMAGE	ATTACK
0.0000	0
43.0064	8
151.7375	16
292.4052	24
443.7391	32
609.1711	40
783.8706	48
968.7585	56
1159.6288	64
1351.7746	72
1546.5210	80
1742.6264	88
1938.0430	96
2087.2653	104
2189.7735	112
2288.9334	120
2384.9613	128
2478.1841	136
2568.1022	144
2654.2757	152
2737.4805	160

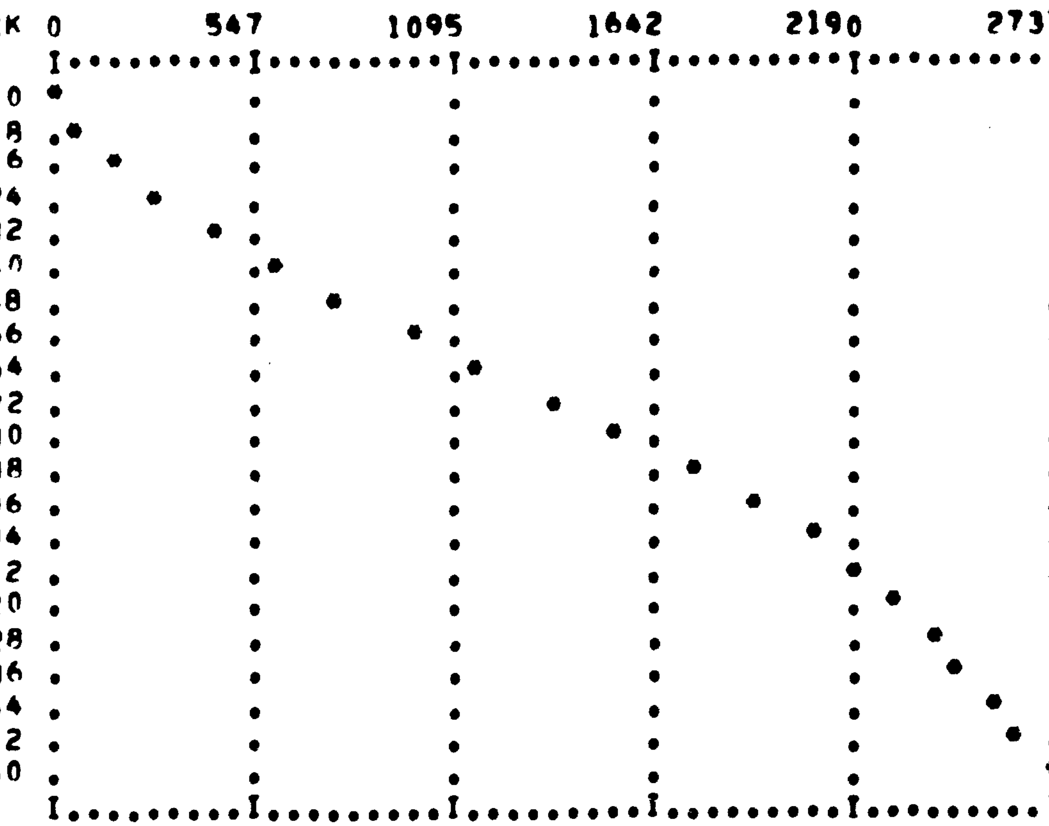

FIGURE 20 CORRIDOR DAMAGE FUNCTIONS

CORRIDOR NUMBER 7
WEAPON NUMBER 2

DAMAGE	ATTACK
0.0000	0
13.7027	3
50.7227	6
106.2572	9
176.9992	12
259.6949	15
351.8597	18
446.3943	21
535.8789	24
627.7335	27
722.9182	30
820.6823	33
919.4912	36
1019.3319	39
1121.2820	42
1226.9143	45
1334.1181	48
1441.0614	51
1547.1497	54

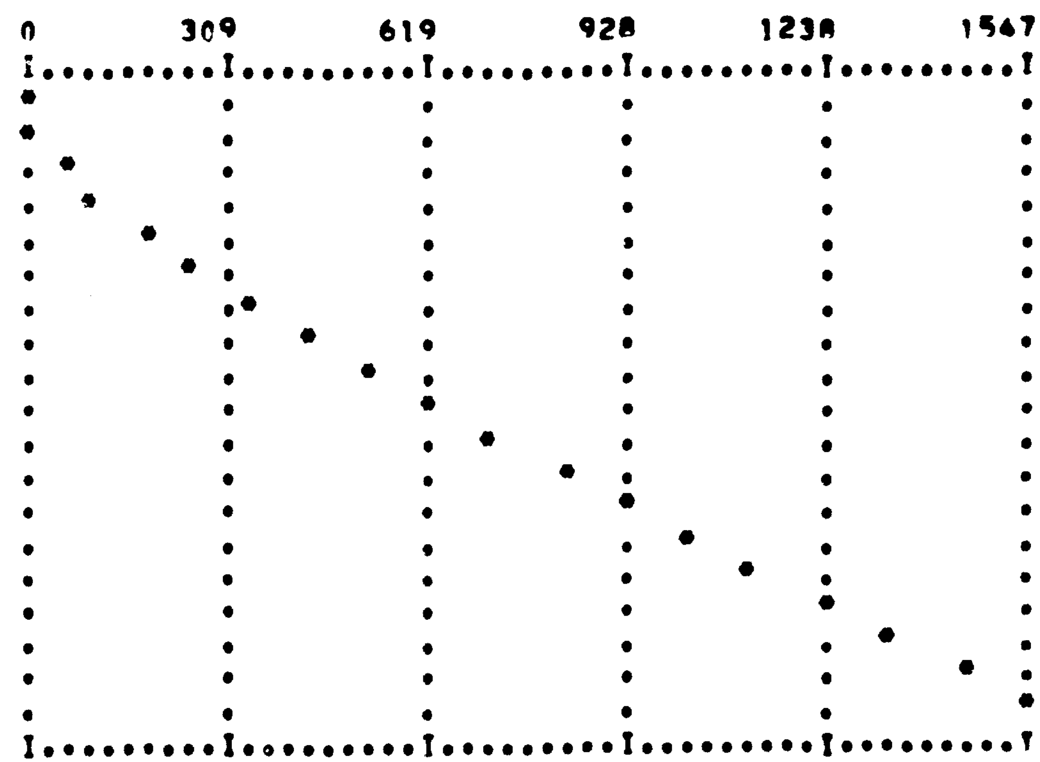

FIGURE 20 CORRIDOR DAMAGE FUNCTIONS (Concluded)

ITERATION NUMBER 8

DX	DLAMBDA	LAMBDA	WEAP DES	WEAP OBT	DAMAGE	LAGRAN
-.1882	-.3137	11.1419	21	63	786.9694	85.0311
-.1882	-.0121	46.5663	37	37	4497.0828	2774.1298

TOTAL ATTACKERS = 100
TOTAL DAMAGE = 5284.0522
TOTAL LAGRANGIAN = 2859.1610
TOTAL VALUE = 110200.00

CORRIC	TYPE	WEAP	DAMAGE
1	0	0	0.00
2	1	21	310.45
3	1	21	236.87
4	0	0	0.00
5	0	0	0.00
6	0	0	0.00
7	1	21	239.65
8	0	0	0.00
9	2	37	4497.08

FIGURE 21 ITERATION SUMMARY

FINAL ALLOCATION TO TARGETS IN CORRIDOR 2
THIS CORRIDOR WAS ATTACKED BY WEAPON 1

TOTAL ATTACKERS TO CORRIDOR = 21
TOTAL DAMAGE TO CORRIDOR = 309.9591
TOTAL VALUE OF CORRIDOR = 11300.00

TARGET	ATTACK	DAMAGE
3	0	0.0000
20	2	26.4272
21	8	120.2752
30	7	109.1215
32	1	13.5355
54	2	28.0465
55	0	0.0000
62	0	0.0000
64	0	0.0000
71	1	12.5532
87	0	0.0000
100	0	0.0000
104	0	0.0000
120	0	0.0000
123	0	0.0000
129	0	0.0000
132	0	0.0000
149	0	0.0000
175	0	0.0000
200	0	0.0000
207	0	0.0000
208	0	0.0000
212	0	0.0000
238	0	0.0000
245	0	0.0000
249	0	0.0000
250	0	0.0000
279	0	0.0000
294	0	0.0000
333	0	0.0000
340	0	0.0000
359	0	0.0000
371	0	0.0000
383	0	0.0000
389	0	0.0000
394	0	0.0000
399	0	0.0000

FIGURE 22 FINAL CORRIDOR ALLOCATION

SUMMARY OF FINAL ALLOCATION

CORRIDOR	WEAP TYP	ATTACK	DAMAGE	VALUE
1	0	0	0.00	3800
2	1	21	309.96	11300
3	0	0	0.00	18920
4	0	0	0.00	23780
5	0	0	0.00	10360
6	0	0	0.00	7280
7	0	0	0.00	5200
8	0	0	0.00	6360
9	2	37	4497.08	23200

WEAP TYP	ATTACK	DAMAGE
1	21	309.9591
2	37	4497.0828

TOTAL ATTACKERS = 58
TOTAL DAMAGE = 4807.0419
TOTAL VALUE = 110200.0000

FIGURE 23 SUMMARY

Appendix

PROGRAM LISTING

```
      PROGRAM BALLOC (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      COMMON/CFDAM/LAMDA,K(500,5),V(500),CRITLM(500,5),IFIRST,ILAST,
     1ATTACK,DAMAGE,ATT(500),DAM(500),NINTER(500,5)
      COMMON/CALLOC/WD(5),LAMBDA(5),ITYPE(15),RWEAP(15),EDAM(15),
     1ATTCRV(15,5,21),DAMCRV(15,5,21),LAMMAX(15,5),INTERV(15),
     2TOTALA(5),TOTALD(5),TOTALL(5),VALUE(15)
      COMMON/CBEST/TWJ,TOTAB(5),ITYPEB(15),RWEAPB(15),EDAMB(15),
     1INTRVB(15),LAMDAB(5),ITERB,ITER
C
      DIMENSION NTARG(15),TDIST(500),IPOINT(15),AP(5),DIST(12),
     1PP(12,5,5),A(21,5),NP(21,5),KAY(500,5),IPRINT(3),WEAPMX(15),
     2IDIST(500),X(21),Y(21),DLAMDA(5),SAM(500,5),LAT(500),
     3LONG(500),CLAT(15,2),CLONG(15,2),KAY1(400,5),REL(5),
     4ID(500),NINCR(5),NINCRP(5),INCR(5),NWEPMX(5),C(5),INTER(5)
C
      REAL KAY,K,LAMDA1,LAMDA2,LAMDA,LAMBDA,LAMMAX,LTOTAL,LAMDAB,
     1LAT,LONG,KAY1
C
      READ 100, NCORID,NATTYP,NAP,NDIST,ITERMX,(IPRINT(I),I=1,3),NEXEMP,
     1NLDEF
      IF (NATTYP.GT.NCORID) GO TO 650
      READ 100, (NINCR(IW),IW=1,NATTYP)
      READ 100, (NWEPMX(IW),IW=1,NATTYP)
      DO 203 IW=1,NATTYP
      NINCRP(IW)=NINCR(IW)+1
  203 INCR(IW)=(FLOAT(NWEPMX(IW))/NINCR(IW)+0.999)
      READ 105, FRACT1,FRACT2,DLLIM
      DO 201 IW=1,NATTYP
  201 READ 105, (KAY1(IE,IW),IE=1,NEXEMP)
      DO 204 IT=1,400
      DO 204 IW=1,NATTYP
  204 NINTER(IT,IW)=0
      DO 206 IL=1,NLDEF
      READ 100, IDT,(INTER(IW),IW=1,NATTYP)
      DO 206 IW=1,NATTYP
  206 NINTER(IDT,IW)=INTER(IW)
      READ 100,   NTARG(IC),IC=1,NCORID)
      IPOINT(1)=1
      DO 200 IC=2,NCORID
  200 IPOINT(IC)=IPOINT(IC-1)+NTARG(IC-1)
      ITARG=IPOINT(NCORID)+NTARG(NCORID)-1
      DO 202 IT=1,ITARG
      READ 106, ID(IT),NEX,V(IT),LAT1,LAT2,LAT3,LONG1,LONG2,LONG3
      LAT(IT)=(LAT1+(LAT2+LAT3/60.)/60.)*0.017453
      LONG(IT)=(LONG1+(LONG2+LONG3/60.)/60.)*0.017453
      IDT=ID(IT)
      DO 202 IW=1,NATTYP
      SAM(IT,IW)=NINTER(IDT,IW)
  202 KAY(IT,IW)=KAY1(NEX,IW)
      DO 205 IC=1,NCORID
      READ 108, CLAT1,CLAT2,CLAT3,CLONG1,CLONG2,CLONG3,CLAT4,CLAT5,CLAT6
     1,CLONG4,CLONG5,CLONG6
      CLAT(IC,1)=(CLAT1+(CLAT2+CLAT3/60.)/60.)*0.017453
```

```
      CLAT(IC,2)=(CLAT4+(CLAT5+CLAT6/60.)/60.)*0.017453
      CLONG(IC,1)=(CLONG1+(CLONG2+CLONG3/60.)/60.)*0.017453
      CLONG(IC,2)=(CLONG4+(CLONG5+CLONG6/60.)/60.)*0.017453
205   CONTINUE
      READ 105, (C(IW),IW=1,NATTYP)
      READ 105, (REL(IW),IW=1,NATTYP)
      VEE=0.0
      DO 215 IC=1,NCORID
      IFIRST=IPOINT(IC)
      ILAST=IFIRST+NTARG(IC)-1
      VALUE(IC)=0.0
      D=ACOS(SIN(CLAT(IC,1))*SIN(CLAT(IC,2))+COS(CLAT(IC,1))*COS(CLAT(IC
     1,2))*COS(CLONG(IC,2)-CLONG(IC,1)))
      GAMMA=ACOS((SIN(CLAT(IC,2))-COS(D)*SIN(CLAT(IC,1)))/(SIN(D)*COS(
     1CLAT(IC,1))))
      DO 212 IT=IFIRST,ILAST
      B=ACOS(SIN(CLAT(IC,1))*SIN(LAT(IT))+COS(CLAT(IC,1))*COS(LAT(IT))*
     1COS(LONG(IT)-CLONG(IC,1)))
      ALPHA=ACOS((SIN(LAT(IT))-COS(B)*SIN(CLAT(IC,1)))/(SIN(B)*COS(CLAT(
     1IC,1))))
      BETA=ALPHA-GAMMA
      TDIST(IT)=3440.*ASIN(SIN(B)*SIN(BETA))
      DO 210 IW=1,NATTYP
      IF (SAM(IT,IW).EQ.0.) GO TO 209
      NINTER(IT,IW)=(SAM(IT,IW)*C(IW)*REL(IW)+0.5)
      GO TO 210
209   NINTER(IT,IW)=0
210   CONTINUE
212   VALUE(IC)=VALUE(IC)+V(IT)
215   VEE=VEE+VALUE(IC)
      DO 221 IC=1,NCORID
      IFIRST=IPOINT(IC)
      ILAST=IFIRST+NTARG(IC)-1
      PRINT 190, IC
      DO 221 IT=IFIRST,ILAST
221   PRINT 195, TD(IT),V(IT),TDIST(IT),(KAY(IT,IW),IW=1,5),(NINTER(IT,
     1IW),IW=1,5)
      PRINT 110
      DO 300 IC=1,NCORID
      IF (IC.EQ.NCORID) GO TO 223
      READ 105, (AP(M),M=1,NAP)
      READ 105, (DIST(I),I=1,NDIST)
      DO 220 IW=1,NATTYP
      DO 220 M=1,NAP
220   READ 105, (PP(I,M,IW),I=1,NDIST)
223   DO 222 IW=1,NATTYP
      ATTCRV(IC,IW,1)=0.0
222   DAMCRV(IC,IW,1)=0.0
      IFIRST=IPOINT(IC)
      ILAST=IFIRST+NTARG(IC)-1
      IF (IC.EQ.NCORID) GO TO 255
      DO 230 IT=IFIRST,ILAST
      DO 225 I=2,NDIST
```

```
      IF (TDIST(IT).GE.DIST(I)) GO TO 225
      IDIST(IT)=I
      GO TO 230
  225 CONTINUE
      IDIST(IT)=NDIST
  230 CONTINUE
      DO 250 IW=1,NATTYP
      MINCRP=NINCRP(IW)
      DO 250 M=2,MINCRP
      A(M,IW)=(M-1)*INCR(IW)
      DO 245 IA=2,NAP
      IF (A(M,IW).GE.AP(IA)) GO TO 245
      NP(M,IW)=IA
      GO TO 250
  245 CONTINUE
      NP(M,IW)=NAP
  250 CONTINUE
  255 DO 300 IW=1,NATTYP
      XLM=0.0
      MINCRP=NINCRP(IW)
      DO 290 M=2,MINCRP
      JB=NP(M,IW)
      JA=JB-1
      CLMAX=0.0
      DO 260 IT=IFIRST,ILAST
      IF (IC.NE.NCORID) GO TO 257
      K(IT,IW)=KAY(IT,IW)
      GO TO 259
  257 KB=IDIST(IT)
      KA=KB-1
      PPROB1=RINT(DIST(KA),DIST(KB),PP(KA,JA,IW),PP(KB,JA,IW),TDIST(IT))
      PPROB2=RINT(DIST(KA),DIST(KB),PP(KA,JB,IW),PP(KB,JB,IW),TDIST(IT))
      PPROB=RINT(AP(JA),AP(JB),PPROB1,PPROB2,A(M,IW))
      IF (PPROB.GT.1.0) PPROB=1.0
      IF (PPROB.LT.0.0) PPROB=0.0
      K(IT,IW)=BRKAY(KAY(IT,IW),PPROB)
      IF (NINTER(IT,IW).EQ.0) GO TO 259
      IF (PPROB.EQ.0.0) GO TO 259
      NINTER(IT,IW)=(NINTER(IT,IW)/PPROB+0.5)
  259 CRITLM(IT,IW)=V(IT)*K(IT,IW)**2/2.
  260 CLMAX=AMAX1(CLMAX,CRITLM(IT,IW))
      LAMDA1=CLMAX
      LAMDA2=0.0
  270 LAMDA=(LAMDA1+LAMDA2)/2.
      IF ((LAMDA1-LAMDA2).LT.0.001) GO TO 283
      CALL FDAM(IW)
      IF (ATTACK-A(M,IW)) 275,287,280
  275 LAMDA1=LAMDA
      GO TO 270
  280 LAMDA2=LAMDA
      GO TO 270
  283 LAMDA=LAMDA1
      CALL FDAM(IW)
      ATEMP=ATTACK
```

```
      DTEMP=DAMAGE
      LAMDA=LAMDA2
      CALL FDAM(IW)
      A1=ABS(ATEMP-A(M,IW))
      A2=ABS(ATTACK-A(M,IW))
      IF (A2.LE.A1) GO TO 285
      ATTACK=ATEMP
      DAMAGE=DTEMP
  285 MO=M-1
      API=A(M,IW)+INCR(IW)
      IF (ATTACK.GT.ATTCRV(IC,IW,MO).AND.ATTACK.LT.API) GO TO 287
      ATTCRV(IC,IW,M)=ATTCRV(IC,IW,MO)
      DAMCRV(IC,IW,M)=DAMCRV(IC,IW,MO)
      GO TO 288
  287 ATTCRV(IC,IW,M)=ATTACK
      DAMCRV(IC,IW,M)=DAMAGE
  288 CONTINUE
      IF (ATTACK.EQ.0.) GO TO 290
      XLM2=DAMAGE/ATTACK
      XLM=AMAX1(XLM,XLM2)
  290 CONTINUE
  291 LAMMAX(IC,IW)=XLM
      IF (IPRINT(1).EQ.0) GO TO 300
      MINCRP=NINCRP(IW)
      DO 298 M=1,MINCRP
      X(M)=ATTCRV(IC,IW,M)
  298 Y(M)=DAMCRV(IC,IW,M)
      PRINT 135, IC,IW
      YMIN=0.0
      YMAX=Y(MINCRP)
      N3=1
      LINE=3
      CALL PLOTN (Y,X,YMIN,YMAX,N3,MINCRP,LINE)
  300 CONTINUE
      CALL SECOND (TIME)
      PRINT 190, TIME
  301 READ 105, (WD(IW),IW=1,NATTYP)
      IF (EOF,5) 999,302
  302 DO 303 IW=1,NATTYP
      IF (WD(IW).LE.NWEPMX(IW)) GO TO 303
      PRINT 199, IW,NWEPMX(IW),WD(IW)
      GO TO 301
  303 CONTINUE
      DO 304 IW=1,NATTYP
      LAMBDA(IW)=0.0
      DO 304 IC=1,NCORID
  304 LAMBDA(IW)=AMAX1(LAMBDA(IW),LAMMAX(IC,IW))
      IFLAG=0
      CALL ALOCAT (NATTYP,NCORID,NINCRP,IFLAG)
      DX=-0.03
      DO 315 IW=1,NATTYP
      IF (TOTALA(IW).LT.WD(IW)) GO TO 305
      DLAMDA(IW)=0.05
```

```
      GO TO 310
  305 DLAMDA(IW)=-0.05
  310 LAMBDA(IW)=LAMBDA(IW)*(1.+DLAMDA(IW))*(1.+DX)
  315 CONTINUE
      ITER=1
      TWJ=10.E40
  320 CALL ALOCAT (NATTYP,NCORID,NINCRP,IPRINT(2))
      CALL RESTCY (NATTYP,NCORID)
      IF (IPRINT(3).EQ.0) GO TO 323
      ATOTAL=0.0
      DTOTAL=0.0
      LTOTAL=0.0
      PRINT 115, ITER
      PRINT 120
      DO 322 IW=1,NATTYP
      ATOTAL=ATOTAL+TOTALA(IW)
      DTOTAL=DTOTAL+TOTALD(IW)
      LTOTAL=LTOTAL+TOTALL(IW)
  322 PRINT 125, DX,DLAMDA(IW),LAMBDA(IW),WD(IW),TOTALA(IW),TOTALD(IW),
     1TOTALL(IW)
      PRINT 130, ATOTAL,DTOTAL,LTOTAL,VEE
  323 DLMAX=0.0
      DO 325 IW=1,NATTYP
  325 DLMAX=AMAX1(DLMAX,ABS(DLAMDA(IW)))
      IF (DLMAX.LT.DLLIM) GO TO 450
      DO 330 IW=1,NATTYP
  330 IF (ABS(TOTALA(IW)-WD(IW)).GE.FLOAT(INCR(IW))) GO TO 335
      DO 332 IW=1,NATTYP
  332 IF (TOTALA(IW).NE.WD(IW)) GO TO 500
      GO TO 505
  335 DO 365 IW=1,NATTYP
      IF (DLAMDA(IW).GT.0.) GO TO 345
      IF (TOTALA(IW).GE.WD(IW)) GO TO 340
      DLAMDA(IW)=FRACT1*DLAMDA(IW)
      GO TO 360
  340 DLAMDA(IW)=FRACT2*DLAMDA(IW)
      GO TO 355
  345 IF (TOTALA(IW).GT.WD(IW)) GO TO 350
      DLAMDA(IW)=FRACT2*DLAMDA(IW)
      GO TO 360
  350 DLAMDA(IW)=FRACT1*DLAMDA(IW)
  355 DLAMDA(IW)=AMAX1(DLMAX/20.,AMIN1(DLAMDA(IW),0.5))
      GO TO 365
  360 DLAMDA(IW)=AMIN1(-DLMAX/20.,AMAX1(DLAMDA(IW),-0.8))
  365 CONTINUE
      TS=0.0
      TA=0.0
      DO 370 IW=1,NATTYP
      TS=TS+LAMBDA(IW)*TOTALA(IW)
  370 TA=TA+LAMBDA(IW)*WD(IW)
      IF (DX.GT.0.) GO TO 380
      IF (TS.GT.TA) GO TO 375
      DX=FRACT1*DX
```

```
      GO TO 395
  375 DX=FRACT2*DX
      GO TO 390
  380 IF (TS.GT.TA) GO TO 385
      DX=FRACT2*DX
      GO TO 395
  385 DX=FRACT1*DX
  390 DX=AMAX1(DLMAX/20.,AMIN1(DX,0.5))
      GO TO 400
  395 DX=AMIN1(-DLMAX/20.,AMAX1(DX,-0.8))
  400 DO 410 IW=1,NATTYP
  410 LAMBDA(IW)=(1.+DLAMDA(IW))*(1.+DX)*LAMBDA(IW)
      ITER=ITER+1
      IF (ITER.GT.ITERMX) GO TO 450
      GO TO 320
  450 DO 460 IW=1,NATTYP
      LAMBDA(IW)=LAMDAB(IW)
  460 TOTALA(IW)=TOTAB(IW)
      DO 470 IC=1,NCORID
      ITYPE(IC)=ITYPEB(IC)
      RWEAP(IC)=RWEAPB(IC)
      EDAM(IC)=EDAMB(IC)
  470 INTERV(IC)=INTRVB(IC)
      ITER=ITERB
  500 CALL CLOSE (NATTYP,NCORID,NINCRP)
  505 DO 510 IW=1,NATTYP
      TOTALD(IW)=0.0
  510 TOTALL(IW)=0.0
      DO 520 IC=1,NCORID
      IW=ITYPE(IC)
      IF (IW.EQ.0) GO TO 520
      TOTALD(IW)=TOTALD(IW)+EDAM(IC)
      TOTALL(IW)=TOTALL(IW)+EDAM(IC)-LAMBDA(IW)*RWEAP(IC)
  520 CONTINUE
      PRINT 140,ITER
      ATOTAL=0.0
      DTOTAL=0.0
      DO 541 IW=1,NATTYP
      TOTALA(IW)=0.0
  541 TOTALD(IW)=0.0
      DO 600 IC=1,NCORID
      IW=ITYPE(IC)
      IF (IW.EQ.0) GO TO 595
      WEAPD=RWEAP(IC)
      CLMAX=0.0
      IFIRST=IPOINT(IC)
      ILAST=IFIRST+NTARG(IC)-1
      IF (IC.EQ.NCORID) GO TO 547
      DO 542 IA=2,NAP
      IF (WEAPD.GE.AP(IA)) GO TO 542
      NPD=IA
      GO TO 543
```

```
542 CONTINUE
    NPD=NAP
543 CONTINUE
    JB=NPD
    JA=JB-1
547 DO 546 IT=IFIRST,ILAST
    IF (IC.EQ.NCORID) GO TO 544
    KB=IDIST(IT)
    KA=KB-1
    PPROB1=RINT(DIST(KA),DIST(KB),PP(KA,JA,IW),PP(KB,JA,IW),TDIST(IT))
    PPROB2=RINT(DIST(KA),DIST(KB),PP(KA,JB,IW),PP(KB,JB,IW),TDIST(IT))
    PPROB=RINT(AP(JA),AP(JB),PPROB1,PPROB2,WEAPD)
    IF (PPROB.GT.1.0) PPROB=1.0
    IF (PPROB.LT.0.0) PPROB=0.0
    K(IT,IW)=ARRAY(KAY(IT,IW),PPROB)
    IF (NINTER(IT,IW).EQ.0) GO TO 544
    IF (PPROB.EQ.0.0) GO TO 544
    NINTER(IT,IW)=(NINTER(IT,IW)/PPROB+0.5)
544 CRITLM(IT,IW)=V(IT)*K(IT,IW)**2/2.
    CLMAX=AMAX1(CLMAX,CRITLM(IT,IW))
546 CONTINUE
    LAMDA1=CLMAX
    LAMDA2=0.0
545 LAMDA=(LAMDA1+LAMDA2)/2.
    CALL FDAM (IW)
    IF (ATTACK-WEAPD) 550,580,555
550 LAMDA1=LAMDA
    GO TO 560
555 LAMDA2=LAMDA
560 IF ((LAMDA1-LAMDA2).LT.0.001) GO TO 570
    GO TO 545
570 CALL CLOSE2 (IW,WEAPD)
580 PRINT 160, IC,IW
    PRINT 162, ATTACK,DAMAGE,VALUE(IC)
    ATOTAL=ATOTAL+ATTACK
    DTOTAL=DTOTAL+DAMAGE
    TOTALA(IW)=TOTALA(IW)+ATTACK
    TOTALD(IW)=TOTALD(IW)+DAMAGE
    EDAM(IC)=DAMAGE
    DO 590 IT=IFIRST,ILAST
590 PRINT 165, ID(IT),ATT(IT),DAM(IT)
    GO TO 600
595 PRINT 170, IC
600 CONTINUE
    PRINT 194
    PRINT 150
    DO 603 IC=1,NCORID
603 PRINT 155, IC,ITYPE(IC),RWEAP(IC),EDAM(IC),VALUE(IC)
    PRINT 193
    DO 605 IW=1,NATTYP
605 PRINT 195, IW,TOTALA(IW),TOTALD(IW)
    PRINT 196, ATOTAL,DTOTAL,VFF
    GO TO 301
```

```
650 PRINT 198
100 FORMAT (16I5)
105 FORMAT (8F10.4)
106 FORMAT (2I10,F10.0,2(3X,I3,2I2),5F5.0)
108 FORMAT (4(3X,F3.0,2F2.0))
110 FORMAT (1H1)
115 FORMAT (1H1,*  ITERATION NUMBER *,I2)
120 FORMAT (//*      DX     DLAMBDA     LAMBDA     WEAP DES   WEAP OBT
   1 DAMAGE        LAGRAN*/)
125 FORMAT (1H ,F6.4,2F10.4,2F10.0,2F12.4)
130 FORMAT (///* TOTAL ATTACKERS = *,F6.0/* TOTAL DAMAGE = *,F10.4/
   1* TOTAL LAGRANGIAN = *,F10.4/* TOTAL VALUE = *,F10.2)
135 FORMAT (////* CORRIDOR NUMBER *,I2/* WEAPON NUMBER *,I2)
140 FORMAT (1H1,*FINAL ALLOCATION FOUND AFTER CLOSING ON ITERATION *,
   1I2)
150 FORMAT (/////* CORRIDOR    WEAP TYP    ATTACK    DAMAGE     VALUE*
   1/)
155 FORMAT (1H ,I5,I12,F12.0,F12.2,F10.0)
160 FORMAT (1H1,*FINAL ALLOCATION TO TARGETS IN CORRIDOR *,I2/* THIS C
   1ORRIDOR WAS ATTACKED BY WEAPON *,I2//)
162 FORMAT (* TOTAL ATTACKERS TO CORRIDOR = *,F5.0/* TOTAL DAMAGE TO C
   1ORRIDOR = *,F12.4/* TOTAL VALUE OF CORRIDOR = *,F10.2//* TARGET
   2    ATTACK       DAMAGE*//)
165 FORMAT (1H ,I5,F12.0,F15.4)
170 FORMAT (1H1,*CORRIDOR *,I2,* WAS NOT ATTACKED*)
180 FORMAT (1H1,*CORRIDOR NUMBER *,I2//*     ID     VALUE     DIST
   1  K*,50X,*INTERCEPTORS*//)
185 FORMAT (1H ,I5,2F10.2,5F10.4,5I8)
190 FORMAT (* TIME = *,F10.4)
193 FORMAT (////* WEAP TYP    ATTACK      DAMAGE*/)
194 FORMAT (1H1,* SUMMARY OF FINAL ALLOCATION*)
195 FORMAT (1H ,I5,F12.0,F13.4)
196 FORMAT (///* TOTAL ATTACKERS = *,F5.0/* TOTAL DAMAGE = *,F12.4/
   1* TOTAL VALUE = *,F12.4)
198 FORMAT (1H1,*ERROR IN INPUT VALUES.*/* NUMBER OF ATTACKER TYPES EX
   1CEEDS THE NUMBER OF CORRIDORS*)
199 FORMAT (1H1,*ERROR IN INPUT.*//* THIS CASE WILL NOT BE RUN.*//
   1* NUMBER OF WEAPONS OF TYPE *,I1,* CANNOT EXCEED *,I5//
   2* THIS CASE REQUESTED *,F5.0,* WEAPONS OF THAT TYPE*)
999 STOP
    END
```

```
      SUBROUTINE F[illegible](IW)
C
      COMMON /C[illegible]/LAMDA,K(500,5),V(500),CRITLM(500,5),IFIRST,ILAST,
     1ATTACK,DAMAGE,ATT(500),DAM(500),NITER(500,5)
C
      REAL LAMDA,K,LAMD1,LAMD2,LAMDAP
C
      ATTACK=0.
      DAMAGE=0.
      DO 300 II=IFIRST,ILAST
      IF (LAMDA.GE.CRITLM(II,IW)) GO TO 280
      ATT=(1./K(II,IW))*ALOG(V(II)*K(II,IW)**2/(2.*LAMDA)))**2
      ATT1=INT(ATT)
      DAM1=DAMGFNC(V(II),K(II,IW),ATT1)
      IF (NITER(II,IW).EQ.0) GO TO 220
      LAMDAP=DAM1/(ATT1*NITER(II,IW))
      IF (LAMDAP.LT.LAMDA) GO TO 280
  220 ATT2=ATT1+1.
      DAM2=DAMGFNC(V(II),K(II,IW),ATT2)
      LAMD1=DAM1-LAMDA*ATT1
      LAMD2=DAM2-LAMDA*ATT2
      IF (LAMD2.GT.LAMD1) GO TO 240
      ATT(II)=ATT1*NITER(II,IW)
      DAM(II)=DAM1
      GO TO 250
  240 ATT(II)=ATT2*NITER(II,IW)
      DAM(II)=DAM2
  250 ATTACK=ATTACK+ATT(II)
      DAMAGE=DAMAGE+DAM(II)
      GO TO 300
  280 ATT(II)=0.
      DAM(II)=0.
  300 CONTINUE
      RETURN
      END



      FUNCTION DAMGFNC(V,K,C)
      DAMGFNC=V*(1.-(1.+K*SQRT(C))*EXP(-K*SQRT(C)))
      RETURN
      END
```

```
      FUNCTION ARRAY(DKAY,RR)
      D=RR*(1.0-EXP(-DKAY)*(1.0+DKAY))
      X=1.0
      IT=1
    5 TEMP=EXP(-X)
      D1=1.0-TEMP*(1.0+X)-D
      IF (ABS(D1) .LT. 1.0E-5) GO TO 10
      IF (IT .GT. 50) GO TO 20
      IT=IT+1
      DP= X*TEMP
      X=X-D1/DP
      GO TO 5
   10 ARRAY=X
      RETURN
   20 PRINT 1000,X,D1
 1000 FORMAT(* ERROR IN ARRAY*,3E12.4)
      RETURN
      END



      FUNCTION FIT(X1,X2,Y1,Y2,X)
C     PERFORMS LINEAR INTERPOLATION
      FIT=Y1-(X1-X)*(Y2-Y1)/(X2-X1)
      RETURN
      END



      SUBROUTINE PLOT (YI,AR,YMIN,YMAX,I3,NPP,LINE)
C     YI,AR ARE ARRAY NAMES
C     NPP = NUMBER OF POINTS PLOTTED
C     CURVE PLOTTED IN POSITIONS 30 TO 80
      DIMENSION YI(1),AR(1),YTICK(11)
      INTEGER SPACE(101),BLANK,DOT
      DATA BLANK,DOT,MARK/1H ,1H.,1H*/
    3 IF (NPP.EQ.0) RETURN
      YM=50./(YMAX-YMIN)
      YTICK(1)=YMIN
      YD10=10./YM
      DO 30 I=1,5
   30 YTICK(I+1)=YTICK(I)+YD10
      PRINT 100, (YTICK(I),I=1,6)
  100 FORMAT (//6X,*DAMAGE*,4X,*ATTACK*,F2.0,5F10.0)
      WRITE (6,104)
 104  FORMAT(29X1HI5(10H.........I))
      NL=I3*NPP
      LI=LINE+1
      DO6I=1,51
```

```
6      SPACE(I)=BLANK
C      SETTING UP GRID LINES
       GO TO (1),11,12,13), L
13     DO 130 I=11,51,10
130    SPACE(I)=DOT
12     SPACE(51)=DOT
11     SPACE(1)=DOT
10     DO 50 NP=1,NL,13
       X=AR(NP)
       MC1=1
       IY1=(Y1(NP)-Y1(1))*YU+1.5
       IF(IY1.LE.0.OR.IY1.GE.52) MC1=0
       IF(MC1.EQ.0) GO TO 1
       ISP1=SPACE(IY1)
       SPACE(IY1)=MARK
1      WRITE(6,103) Y1(NP),X,(SPACE(I), I = 1,51)
103    FORMAT(2X,F13.4,F13.0,1X51A1)
49     IF(MC1.NE.0) SPACE(IY1)=ISP1
50     CONTINUE
       WRITE (6, 104)
       RETURN
       END



       SUBROUTINE ALOCAT (NATTYP,NCORID,NINCRP,IFLAG)
C
       COMMON/CALLOC/WT(5),LAMBDA(5),ITYPE(15),RWEAP(15),EDAM(15),
      1ATTCRV(15,5,21),DAMCRV(15,5,21),LAMMAX(15,5),INTERV(15),
      2TOTALA(5),TOTALD(5),TOTALL(5),VALUE(15)
C
       DIMENSION NINCRP(5)
C
       REAL LAMBDA,LAMMAX
C
       DO 200 I=1,NATTYP
       TOTALA(I)=0.0
       TOTALD(I)=0.0
  200  TOTALL(I)=0.0
       DO 300 IC=1,NCORID
       ITYPE(IC)=0
       RWEAP(IC)=0.0
       EDAM(IC)=0.0
       INTERV(IC)=0
       WEAPS=0.0
       DO 290 IW=1,NATTYP
       IF (LAMBDA(IW).GT.LAMMAX(IC,IW)) GO TO 290
       RLAM1=0.0
       MINCRP=NINCRP(IW)
       DO 210 K=2,MINCRP
       RLAG=DAMCRV(IC,IW,K)-LAMBDA(IW)*ATTCRV(IC,IW,K)
```

```
      IF (HLAS.LE.HLAM1) GO TO 210
      HLAM1=HLAS
      KK=K
  210 CONTINUE
      HA=ATTCRV(IC,IW,KK)
      HD=DAMCRV(IC,IW,KK)
      IF (HA.LE.W(IW)) GO TO 250
      HA=W(IW)
      DO 220 K=2,MINCHP
      IF (HA.GT.ATTCRV(IC,IW,K)) GO TO 220
      KK=K
      GO TO 230
  220 CONTINUE
      KK=MINCHP
  230 KJ=KK-1
      HD=RINT(ATTCRV(IC,IW,KJ),ATTCRV(IC,IW,KK),DAMCRV(IC,IW,KJ),
     1DAMCRV(IC,IW,KK),HA)
      HLAM1=HD-LAMBDA(IW)*HA
  250 IF (HLAM1.LE.HMAXS) GO TO 290
      ITYPE(IC)=IW
      WEAP(IC)=HA
      EDAM(IC)=HD
      IATCRV(IC)=KK
      HMAXS=HLAM1
  290 CONTINUE
      IW=ITYPE(IC)
      TOTALA(IW)=TOTALA(IW)+WEAP(IC)
      TOTALD(IW)=TOTALD(IW)+EDAM(IC)
      TOTALL(IW)=TOTALL(IW)+HMAXS
  300 CONTINUE
      IF (IFLAG.EQ.0) GO TO 999
      PRINT 100
      DO 310 IC=1,NCHMIN
  310 PRINT 110, IC,ITYPE(IC),WEAP(IC),EDAM(IC)
  100 FORMAT (1H1,*CHMNT      TYPE       WEAP        DAMAGE*/)
  110 FORMAT (I5,I10,I10,F12.2,F12.2)
  999 RETURN
      END
```

```
      SUBROUTINE BESTEV(NATTYP,NCOMID)
C
      COMMON/CALLBC/WD(5),LAMBDA(5),ITYPE(15),RWEAP(15),EDAM(15),
     1ATTCRA(15,5,21),DAMCRV(15,5,21),LAMMAX(15,5),INTERV(15),
     2TOTALA(5),TOTALD(5),TOTALL(5),VALUE(15)
      COMMON/CBEST/TWJB,TOTAB(5),ITYPEB(15),RWEAPB(15),EDAMB(15),
     1INTRVB(15),LAMDAB(5),ITERB,ITER
C
      REAL LAMBDA,LAMDAB
C
C
      TWJ=0.0
      DO 250 JJ=1,NATTYP
      IF (WD(JJ).EQ.0.0) GO TO 200
      TWJ=TWJ+((TOTALA(JJ)-WD(JJ))/WD(JJ))**2
      GO TO 250
  200 TWJ=TWJ+TOTALA(JJ)**2
  250 CONTINUE
      IF (TWJ.GT.TWJB) GO TO 500
      TWJB=TWJ
      DO 400 JJ=1,NATTYP
      LAMDAB(JJ)=LAMBDA(JJ)
  400 TOTAB(JJ)=TOTALA(JJ)
      DO 450 JJ=1,NCOMID
      ITYPEB(JJ)=ITYPE(JJ)
      RWEAPB(JJ)=RWEAP(JJ)
      EDAMB(JJ)=EDAM(JJ)
  450 INTRVB(JJ)=INTERV(JJ)
      ITERB=ITER
  500 RETURN
      END
```

```
      SUBROUTINE CLOSE (NATTYP,NCORID,NINCRP)
C
      COMMON/CALLOC/WD(5),LAMBDA(5),ITYPE(15),RWEAP(15),EDAM(15),
     1ATTCRV(15,5,21),DAMCRV(15,5,21),LAMMAX(15,5),INTERV(15),
     2TOTALA(5),TOTALD(5),TOTALL(5),VALUE(15)
C
      DIMENSION NINCRP(5)
C
      REAL LAMBDA
C
C
      DO 300 IW=1,NATTYP
      IF (TOTALA(IW).LE.WD(IW)) GO TO 300
      DEC=TOTALA(IW)-WD(IW)
  200 SLOPE=10.E40
      DO 250 IC=1,NCORID
      IF (ITYPE(IC).NE.IW) GO TO 250
  210 IF (INTERV(IC).EQ.1) GO TO 250
      KK=INTERV(IC)
      KJ=KK-1
      IF (ATTCRV(IC,IW,KK).NE.ATTCRV(IC,IW,KJ)) GO TO 220
      INTERV(IC)=INTERV(IC)-1
      GO TO 210
  220 SLOPE1=(DAMCRV(IC,IW,KK)-DAMCRV(IC,IW,KJ))/(ATTCRV(IC,IW,KK)-
     1ATTCRV(IC,IW,KJ))
      IF (SLOPE1.GT.SLOPE) GO TO 250
      ICDECR=IC
  250 CONTINUE
      IC=ICDECR
      KK=INTERV(IC)
      KJ=KK-1
      DIFF=RWEAP(IC)-ATTCRV(IC,IW,KJ)
      IF (DIFF.GE.DEC) GO TO 260
      TOTALA(IW)=TOTALA(IW)-DIFF
      RWEAP(IC)=ATTCRV(IC,IW,KJ)
      EDAM(IC)=DAMCRV(IC,IW,KJ)
      IF (RWEAP(IC).EQ.0.0) ITYPE(IC)=0
      INTERV(IC)=INTERV(IC)-1
      DEC=DEC-DIFF
      GO TO 200
  260 RWEAP(IC)=RWEAP(IC)-DEC
      EDAM(IC)=RINT(ATTCRV(IC,IW,KJ),ATTCRV(IC,IW,KK),DAMCRV(IC,IW,KJ),
     1DAMCRV(IC,IW,KK),RWEAP(IC))
      TOTALA(IW)=TOTALA(IW)-DEC
  300 CONTINUE
      DO 400 IW=1,NATTYP
      MINCRP=NINCRP(IW)
      IF (TOTALA(IW).GE.WD(IW)) GO TO 400
      IF (TOTALA(IW).EQ.0.) GO TO 400
      XINC=WD(IW)-TOTALA(IW)
  310 SLOPE=0.0
      ICINCR=0
```

```
      DO 350 IC=1,NCORID
      IF (ITYPE(IC).NE.IW) GO TO 350
  320 IF (INTERV(IC).EQ.MINCRP) GO TO 350
      KK=INTERV(IC)
      KL=KK+1
      IF (ATTCRV(IC,IW,KL).NE.ATTCRV(IC,IW,KK)) GO TO 330
      INTERV(IC)=INTERV(IC)+1
      GO TO 320
  330 SLOPE1=(DAMCRV(IC,IW,KL)-DAMCRV(IC,IW,KK))/(ATTCRV(IC,IW,KL)-
     1ATTCRV(IC,IW,KK))
      IF (SLOPE1.LT.SLOPE) GO TO 350
      SLOPE=SLOPE1
      ICINCR=IC
  350 CONTINUE
      IF (ICINCR.NE.0) GO TO 370
      DO 360 IC=1,NCORID
      IF (ITYPE(IC).NE.IW) GO TO 360
      EXTRA=WD(IW)-TOTALA(IW)
      RWEAP(IC)=RWEAP(IC)+EXTRA
      TOTALA(IW)=TOTALA(IW)+EXTRA
      NN=MINCRP
  352 NM=NN-1
      IF (ATTCRV(IC,IW,NM).NE.ATTCRV(IC,IW,NN)) GO TO 355
      NN=NM
      GO TO 352
  355 EDAM(IC)=RINT(ATTCRV(IC,IW,NM),ATTCRV(IC,IW,MINCRP),DAMCRV(IC,IW,
     1NM),DAMCRV(IC,IW,MINCRP),RWEAP(IC))
      IF (EDAM(IC).GT.VALUE(IC)) EDAM(IC)=VALUE(IC)
      GO TO 400
  360 CONTINUE
  370 IC=ICINCR
      KK=INTERV(IC)
      KL=KK+1
      DIFF=ATTCRV(IC,IW,KL)-RWEAP(IC)
      IF (DIFF.GE.XINC) GO TO 390
      TOTALA(IW)=TOTALA(IW)+DIFF
      RWEAP(IC)=ATTCRV(IC,IW,KL)
      EDAM(IC)=DAMCRV(IC,IW,KL)
      INTERV(IC)=INTERV(IC)+1
      XINC=XINC-DIFF
      GO TO 310
  390 RWEAP(IC)=RWEAP(IC)+XINC
      EDAM(IC)=RINT(ATTCRV(IC,IW,KK),ATTCRV(IC,IW,KL),DAMCRV(IC,IW,KK),
     1DAMCRV(IC,IW,KL),RWEAP(IC))
      TOTALA(IW)=TOTALA(IW)+XINC
  400 CONTINUE
      DO 450 IW=1,NATTYP
      MINCRP=NINCRP(IW)
      IF (TOTALA(IW).NE.0.) GO TO 450
      ICBEST=0
      DAMB=0.0
      DO 440 IC=1,NCORID
```

```
      IF (ITYPE(IC).NE.0) GO TO 440
      DO 420 K=2,NINCRP
      IF (WD(IW).GT.ATTCRV(IC,IW,K)) GO TO 420
      KK=K
      GO TO 430
420   CONTINUE
      KK=NINCRP
425   KJ=KK-1
      IF (ATTCRV(IC,IW,KJ).NE.ATTCRV(IC,IW,KK)) GO TO 430
      KK=KJ
      GO TO 425
430   KJ=KK-1
      DAMB1=RINT(ATTCRV(IC,IW,KJ),ATTCRV(IC,IW,KK),DAMCRV(IC,IW,KJ),
     1DAMCRV(IC,IW,KK),WD(IW))
      IF (DAMB1.GT.VALUE(IC)) DAMB1=VALUE(IC)
      IF (DAMB1.LT.DAMB) GO TO 440
      DAMB=DAMB1
      ICBEST=IC
440   CONTINUE
      IC=ICBEST
      RNEWP(IC)=WD(IW)
      EDAM(IC)=DAMB
      ITYPE(IC)=IW
      TOTALA(IW)=WD(IW)
450   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CLOSE2 (IW,WEAPO)
C
      COMMON/CRDAM/LAMDA,K(500,5),V(500),CRITLM(500,5),IFIRST,ILAST,
     1ATTACK,DAMAGE,ATT(500),DAM(500),NINTER(500,5)
C
      REAL LAMDA,K,LAMBDA,LAMDAK
C
C
      IF (ATTACK.GT.WEAPO) GO TO 300
      XINC=WEAPO-ATTACK
  200 IF (XINC.EQ.0.) GO TO 500
      SLOPE=0.0
      ITBEST=0
      DO 250 IT=IFIRST,ILAST
      IF (ATT(IT).EQ.0..AND.NINTER(IT,IW).NE.0) GO TO 250
      ATT1=ATT(IT)
      ATT2=ATT1+1.
      DAM1=DAM(IT)
      DAM2=DMGFNC(V(IT),K(IT,IW),ATT2)
      SLOPE1=DAM2-DAM1
      IF (SLOPE1.LE.SLOPE) GO TO 250
      SLOPE=SLOPE1
      ITBEST=IT
  250 CONTINUE
      ATT(ITBEST)=ATT(ITBEST)+1.
      DAMTMP=DMGFNC(V(ITBEST),K(ITBEST,IW),ATT(ITBEST))
      DAMINC=DAMTMP-DAM(ITBEST)
      DAM(ITBEST)=DAMTMP
      ATTACK=ATTACK+1.
      DAMAGE=DAMAGE+DAMINC
      XINC=XINC-1.
      GO TO 200
  300 XDEC=ATTACK-WEAPO
  310 IF (XDEC.EQ.0.) GO TO 500
      SLOPE=10.E+0
      ITBEST=0
      DO 350 IT=IFIRST,ILAST
      IF (ATT(IT).EQ.0.) GO TO 350
      IF (NINTER(IT,IW).NE.0) GO TO 350
      ATT1=ATT(IT)
      ATT2=ATT1-1.
      DAM1=DAM(IT)
      DAM2=DMGFNC(V(IT),K(IT,IW),ATT2)
      SLOPE1=DAM1-DAM2
      IF (SLOPE1.GT.SLOPE) GO TO 350
      SLOPE=SLOPE1
      ITBEST=IT
  350 CONTINUE
      ATT(ITBEST)=ATT(ITBEST)-1.
      DAMTMP=DMGFNC(V(ITBEST),K(ITBEST,IW),ATT(ITBEST))
      DAMDEC=DAM(ITBEST)-DAMTMP
      DAM(ITBEST)=DAMTMP
```

```
      ATTACK=ATTACK-1.
      DAMAGE=DAMAGE-DAMDEC
      XDEC=XDEC-1.
      GO TO 310
  500 RETURN
      END
```

REFERENCES

1. R. de Sobrino, "Strategic Bomber Effectiveness Evaluation Models; Some General Considerations," Prepared for Defense Communications Agency, SED RM-294, SRI Project 8262 (May 1970).

2. R. de Sobrino, et al., "Development of Techniques Applicable to the Improved Representation of Manned Aircraft Systems in Strategic Systems Effectiveness Evaluation Models," SED-RM-299, SRI Project 8262, Stanford Research Institute, Menlo Park, California (June 1971).

3. I. Weinstein, "A Damage Function for Targets Defended by Both Area and Local Defenses," Technical Note SED-TN-186, SRI Project 5188, Stanford Research Institute, Menlo Park, California (August 1967).

4. R. de Sobrino, G. G. Furman, and E. A. Torbett, "Code 50 (Lambda Corporation) and OPSTRA (Stanford Research Institute): A Comparison of Two Strategic Force Evaluation Models (U)," SRI-SED-RM-5205-53, SRI Project 5205, Stanford Research Institute, Menlo Park, California (1969), SECRET.

5. G. G. Furman, "An Introduction to Program OPSTRA3 and OPSTRA5 and a Comparison of all Five OPSTRA-Series Programs," SED-TN-474, SRI Project 8974, Stanford Research Institute, Menlo Park, California (November 1970).

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Stanford Research Institute Menlo Park, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP N A

3. REPORT TITLE

PROGRAM BALLOC: A MIXED-WEAPON ALLOCATION MODEL AGAINST MULTIPLE NON-OVERLAPPING AREA DEFENSES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Special Report Covering the period 15 August to 15 October 1971

5. AUTHOR(S) *(Last name, first name, initial)*

Ricardo de Sobrino Barbara J. Ripple Nancy J. Lemons

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1971	65	5

8a. CONTRACT OR GRANT NO. Contract N00014-71-C-0015	9a. ORIGINATOR'S REPORT NUMBER(S) Special Report SRI Project 8777
b. PROJECT AND TASK NO.	
c. DOD ELEMENT	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d. DOD SUBELEMENT	

10. AVAILABILITY/LIMITATION NOTICES

Reproduction in whole or in part is permitted for any purpose of the United States government.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Department of the Navy Arlington, Virginia 222 7

13. ABSTRACT

Program BALLOC is a model designed to optimally allocate a mixed-weapon offensive force over a set of multiple-valued targets defended by multiple nonintersecting random area and subtractive local defenses.

The program will maximize the expected damage over the total target set, with the restriction that no more than one weapon type be allocated to any randomly defended island or target subset.

This report presents the concept on which the model is based, and describes the implementation of this concept as a computer program. It also contains a users manual, and an example of the use of the model, employing fictitious data to avoid the necessity of security classification.

DD FORM 1 JAN 64 1473

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Strategic effectiveness evaluation models						
Airborne systems						
Offense						
Defense						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.